

Remarks

Claims 9, 30, 41 and 42 have been cancelled without prejudice. Claims 1, 6, 10-15, 22, 27, 31-36, 39, 40 and 43 have been amended.

The Examiner has rejected applicant's claims 1-6, 9-16, 19-27, 30-37 and 39-43 under 35 U.S.C. §103(a) as being unpatentable over the Patton, et al. (US 6,408,301) patent in view of the Jernigan, IV, et al. (US 5,574,907) patent. The Examiner has also rejected applicant's claims 17-18 under 35 U.S.C. §103(a) as being unpatentable over the Patton, et al. patent in view of the Jernigan, IV, et al. patent further in view of the Levy, et al. (US 6,505,160) patent.

Applicant has cancelled claims 9, 30, 41 and 42, thereby obviating the Examiner's rejection with respect to these claims. Applicant has amended applicant's independent claims 1, 22, 39, 40 and 43, and with respect to such claims, as amended, and their respective dependent claims, the Examiner's rejections are respectfully traversed.

Applicant's independent claims 1, 22, 39, 40 and 43 have been amended to better define applicant's invention. More particularly, applicant's independent claim 1 recites an information processing method for storing a plurality of files having both binary data and metadata related to the binary data into a storage medium in which: a first storage step stores the metadata of the plurality of files into a first block storage area that is a continuous area capable of storing metadata of the plurality of files on said storage medium; a second storage step stores binary data of the plurality of files related to the metadata into a second block storage area other than the first block storage area on said storage medium; and a third storage step stores link information that links the metadata of the plurality of files stored in the first block storage area with the binary data of the plurality of files stored in the second block storage area, in correspondence with the metadata, into the first block storage area, wherein

at the third storage step, each of the link information is stored into an area adjacent to an area where corresponding metadata is stored. Applicant's independent claims 22, 39, 40 and 43 have been similarly amended.

Such a construction is not taught or suggested by the cited art of record. In particular, the Examiner has argued as follows: "Patton et al discloses . . . 'an allocation step of allocating a first block storage that is a continuous area capable of storing metadata of the plurality of files' . . ."; "[w]hen Patton stores the metadata . . . this area corresponds to the 'first storage area'; "[t]he 'second storage' corresponds to the area that stored the image data"; "[t]he 'link information' corresponds to the 'image links'; "[t]he third storage' corresponds the area that stored the image links"; "[l]ink information is stored into an area adjacent to an area where said metadata is stored"; "[t]he 'image link' is derived from the metadata, and is stored in the disc 16 in Fig. 3"; and "[t]herefore, the 'link information' must be stored adjacent with the metadata." The Examiner cites column 2, lines 23-35, column 3, lines 43-60, column 4, lines 20-28 and 39-47 and FIGS. 1 and 3 of the Patton, et al. patent to support the Examiner's argument.

Applicant respectfully disagrees. In the first place, none of the passages cited by the Examiner, disclose or suggest that the link information has to be stored into an area adjacent to an area where corresponding metadata is stored. FIG. 3 cited by the Examiner, shows a Digital Video Disk 16 on which data can be stored, but neither the figure nor the description in the Patton, et al. patent teach or suggest that link information be stored adjacent the corresponding metadata. The fact that the link information may be derived from the metadata or that the both types of data may be on the same disk, simply is not enough to conclude that

they "must be stored adjacent," as the Examiner has argued. To conclude otherwise would give no meaning to the word adjacent.

Moreover, it is evident from the Patton, et al. patent that there is no teaching or suggestion of storing metadata in a storage area that is a continuous area capable of storing metadata of a plurality of files. The Examiner has acknowledged this but has cited the Jernigan, IV, et al. patent and argued that the teachings of this patent would suggest implementing such a feature in the system of the Patton, et al. patent.

Applicant again respectfully disagrees. The Examiner has pointed out that the Jernigan, IV, et al. patent discloses a method for defragmenting already stored file data requiring two stages, where the first stage rearranges the File allocation Table (FAT) and Microsoft DoubleSpace File Allocation Table (MDFAT) entries into adjacent clusters and the second stage moves the data into adjacent variable length clusters such that the data is rearranged in adjacent sectors with no intervening vacant sectors (col. 8, lines 44-49).

Specifically, column 4, lines 38-44 and column 8, lines 44-49 of the Jernigan, IV, et al. patent disclose a defragmentation process requiring two passes such that a file is arranged to be contiguous. In the first pass, the defragmentation process rearranges the already stored entries of a file in the FAT and MDFAT into adjacent clusters, while in the second pass, already stored data is moved corresponding to these entries into adjacent clusters so that the data is rearranged with no intervening vacant sectors. Col. 8, lines 44-49.

The Jernigan, IV, et al. patent thus deals with already stored data and with reconfiguring the already stored data and not with the storing of data. Moreover, the patent teaches rearranging the already stored FAT and MDFAT entries of a file into adjacent clusters, and then moving data of the file corresponding to these entries into adjacent clusters

so that the data is rearranged to have no intervening vacant sectors. Accordingly, nothing is taught or suggested regarding the storing of metadata, let alone the storing of metadata of a plurality of files in a continuous area capable of storing the metadata of such plurality of files.

In point of fact, a skilled artisan viewing the Patton, et al. patent and the Jernigan, IV, et al. patent might, at most, be motivated to reconfigure the already stored data based on the Jernigan IV teachings regarding FAT and MDFAT for a file. However, the skilled artisan would not be motivated to store data differently, nor would the artisan be motivated to store the metadata for a plurality of files in a continuous area capable of storing the metadata of such plurality of files.

Finally, the Examiner has acknowledged that the Patton, et al. patent does not specifically disclose the order of storing the binary data, metadata, and linking data. The Examiner has argued that one of ordinary skill in the art would have recognized that either the metadata or the binary data might have been stored first, that the choice of sequence provides no unexpected or unobvious result and that the ordinary skilled artisan would have recognized that the linking of metadata and binary data would have to occur after those two types of data have been captured or stored. However, the Examiner has not cited any pertinent art in support of this argument. Applicant disagrees with the Examiner's argument and respectfully requests that the Examiner provide prior art supporting this argument.

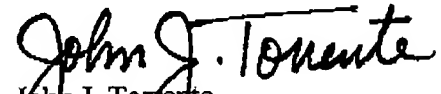
For all the above reasons, applicant submits that applicant's independent claims 1, 22, 39, 40 and 43, and their respective dependent claims, thus patentably distinguish over the combination of the Patton, et al. and the Jernigan, IV, et al. patents. Moreover, there is nothing taught or suggested in the Levy, et al. patent to change this conclusion.

In view of the above, it is submitted that applicant's claims, as amended, patentably distinguish over the cited art of record. Accordingly, reconsideration of the claims is respectfully requested.

Dated: December 8, 2004

COWAN, LIEBOWITZ & LATMAN, P.C.
Avenue of the Americas
New York, NY 10036
(212) 790-9273

Respectfully submitted,


John J. Torrente
Reg. No. 26,359
Attorney for Applicant